Noname manuscript No.
(will be inserted by the editor)

A genetic approach to Markovian characterisation of H.264 scalable video

Dieter Fiems · Bart Steyaert · Herwig Bruneel

Received: date / Accepted: date

Abstract We propose an algorithm for multivariate Markovian characterisation of H.264/SVC scalable video traces at the sub-GoP (Group of Pictures) level. A genetic algorithm yields Markov models with limited state space that accurately capture temporal and inter-layer correlation. Key to our approach is the covariance-based fitness function. In comparison with the classical Expectation Maximisation algorithm, ours is capable of matching the second order statistics more accurately at the cost of less accuracy in matching the histograms of the trace. Moreover, a simulation study shows that our approach outperforms Expectation Maximisation in predicting performance of video streaming in various networking scenarios.

Keywords H.264/SVC · traffic characterisation · Markovian arrival process

1 Introduction

A video stream is called scalable if parts of the stream (layers) can be removed such that the resulting substream forms another valid video stream for some target decoder and such that this substream represents the source content at a lower quality in comparison with the original stream [1–3]. Prime benefits of efficient scalable video coding for video transmission over packet switched networks include support for heterogeneous clients in multicast transmission scenarios and the capability of graceful degradation of the video quality whenever this is required by the network conditions. Since a single stream offers the same content in different formats, heterogeneous clients may decode multicast video streams in their preferred version: preferred spatial resolution, temporal resolution, or preferred fidelity. Moreover, by service differentiation, the network can ensure delivery of some substreams of the video stream while discarding other substreams during network congestion. From the vantage point of the receivers, the perceived video quality then degrades gracefully during congestion: the spatial or temporal resolution or the fidelity are temporarily reduced. Although video coding

D. Fiems, B. Steyaert, H. Bruneel
SMACS Research Group, Department of Telecommunications and Information Processing, Ghent University, St-Pietersniewstraat 41, 9000 Gent, Belgium. E-mail: {df,bs,hb}@telin.UGent.be

standards such as MPEG-2, H.263 and MPEG-4 Visual already support various scalability modes, those have been rarely used in practice due to decreased coding efficiency and increased coding complexity, issues which the Scalable Video Coding extension (H.264/SVC) of H.264/AVC now addresses [3].

We consider statistical characterisation of H.264 scalable video coding streams (H.264/SVC). Sufficiently detailed statistical traffic models are important for network design and performance evaluation [4]. A good statistical model is one which can predict one or more performance measures of the real video source in various networking scenarios. It should lead to comparable performance predictions when used instead of the actual traces. In addition to accuracy, Liebeherr [5] mentions speed of the characterisation algorithm and simplicity of the stochastic model as desirable properties. A stochastic model is simple if generation of traces of the model is easy and fast. Finally, if the model is to be used as input for various performance or queueing analyses, the mathematical tractability of the associated performance or queueing models is also required. Obviously, there is a trade-off between the mathematical tractability, and the accuracy of the traffic models.

Characterisation of variable bit rate video has been an active research area for almost 20 years; see a.o. the survey by Izquierdo and Reeves [6] on statistical traffic models for MPEG-1 encoded video. Proposed statistical models include Markovian processes [7–12], autoregressive processes [13–16], spatial renewal processes [17] and self-similar processes [18, 19]. From the vantage point of queueing theory, Markovian processes are the most tractable: the associated queueing behaviour can be investigated efficiently by means of matrix analytic methods; see for example [20, 21]. Markovian characterisation of video is also the subject of the present paper. Therefore we first survey related literature on Markovian modelling of video streams in more detail.

Given some trace — that is, given the sizes of the consecutive frames or of the consecutive groups of pictures (GoP) — many authors construct Markov processes by quantisation of the frame or GoP sizes: the size of a frame or GoP is mapped on a Markovian state. Conti et al. [7] and Conti and Gregori [9] make use of such a quantisation for the Markovian characterisation of MPEG video. The authors in [7] obtain a Markov chain for the GoPs of an MPEG-1 encoded video by joint quantisation of the GoP sizes and a moving average of the GoP sizes. This study is then extended to include characterisation at the frame-level in [9]. In the latter paper, the authors compare encoding at different quality levels and show that traces at one quality level can be approximated by applying a deterministic mapping function on the corresponding traces at another quality level. Baey [12] constructs a Discrete Batch Markovian Arrival Process (DBMAP) for MPEG-4 video by a scene detection algorithm which depends on quantisation levels. The author considers video traffic at the sub-GoP level but interlayer correlation is not taken into account. All of the preceding papers consider single-layer traces. The quantisation approach also extends to traces with multiple layers. This is e.g. the case for scalable video. Now, the trace specifies the frame size at each layer; each "sample point" of the trace is a vector whose elements correspond to the frame (or GoP) sizes at the different layers. For an MPEG-4 trace, Zhao and Ahmad [11] use a clustering algorithm to partition the set of (vector-valued) frame sizes of the trace and associate states of the Markov process with the different partitions.

Although a quantisation approach yields Markov processes which capture the sample size distribution accurately, direct mapping from sizes to states does not always yield processes that accurately capture the autocorrelation of the trace. Therefore, some authors no longer apply such a direct mapping and the states of the Markov

chain no longer have a simple interpretation. A state is then an abstract concept of the Markov process which is deemed suitable for a trace if it accurately captures a number of statistics of this trace. Lombardo et al. [8] construct a periodic Markov chain to account for the periodicity which is induced by the GoP structure of MPEG traces. A genetic algorithm is used to solve the inverse eigenvalue problem which relates the approximation of the autocorrelation function of the trace to the state transition probabilities of the Markov process. Moltchanov et al. [10] avoid this inverse eigenvalue problem by considering Markov chains with additional structure. Finally, Kempken and Luther [22] consider single-layer H.264 video at the GoP level and use a genetic algorithm to assign states to the GoPs such that the statistics of the model accurately approximate those of the trace. These authors show that a genetic approach yields Markovian models with small state spaces that can capture the time correlation of the trace accurately.

We here extend the approach of [22] to Markovian characterisation of multi-layered video at the sub-GoP level. In particular, the different types of frames are grouped into a number of traffic classes and the H.264/SVC source is investigated at the traffic class level. As indicated above, only few statistical traffic models focus on the scalability of the video sources. Existing models describe two layers at most, a base layer and a (temporal) enhancement layer. However, traffic models for H.264/SVC video need to take into account the correlation between the base and enhancement layers not only in the temporal, but also in the spatial and quality scalability dimensions. Capturing the multi-layered structure of H.264/SVC is the subject of this paper. Key to our approach is the introduction of the matrix-valued covariance function as a second order statistic by which the fitness of candidate Markov models is assessed.

The remainder of this paper is organised as follows. In the next section, the H.264/SVC traces and their statistical properties are introduced. We then focus on the properties of multivariate Markovian processes in section 3. The genetic algorithm for video characterisation is presented in section 4 and numerically evaluated in section 5. To benchmark our approach, we compare the Markov processes generated by our approach with those generated by the well known expectation maximisation (EM) approach [27, 31]. Finally conclusions are drawn in section 6.

2 H.264/SVC traces

To investigate the Markov characterisation algorithm described further, we have two traces at our disposal. These H.264/SVC traces are encoded with the SVC reference software version JSVM 8.9. The traces "Brotherhood of the Wolf" and "Pirates of the Caribbean" consist of 13447 GoPs and 12770 GoPs, respectively. Both traces are encoded into a base layer and a Medium Grain Scalability (MGS) quality enhancement layer. Within each of these layers there are 5 temporal layers (TL) as depicted in Figure 1. Both the base and enhancement layer consist of intra-coded (I), inter-coded (P) and bidirectional coded (B) frames. The arrows indicate decoding dependencies of the different frames and the subscripts indicate the temporal level (temp. level) of the frames. Apart from the indicated decoding dependencies, decoding the frames of the enhancement layer requires the corresponding frames of the base layer. Moreover, there are also dependencies between GoPs. For H.264/SVC, a GoP is a set of pictures between two successive frames of the temporal base layer together with the succeeding

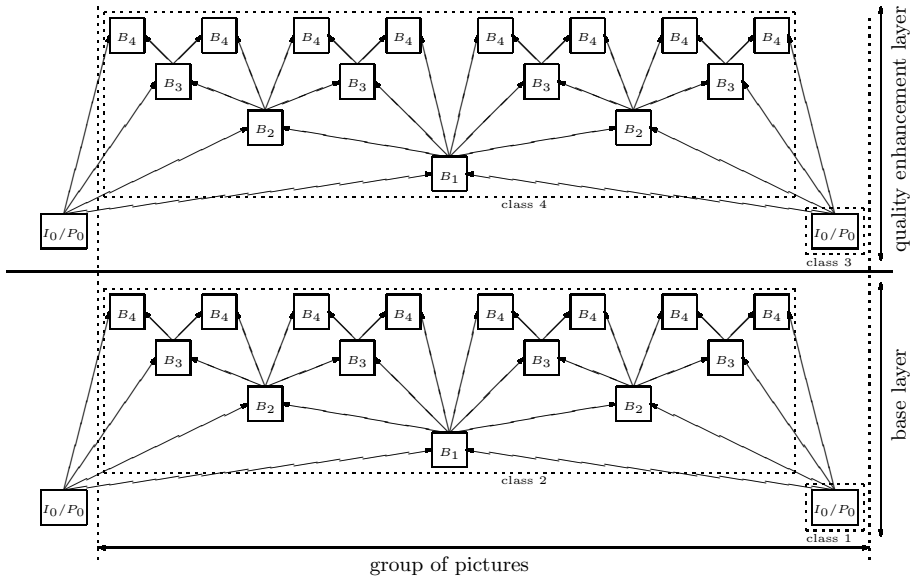

Fig. 1 Group of picture structure of the H.264 traces under consideration.

base layer frame such that GoPs do not necessarily end in an I-frame [3]. For the traces at our disposal, GoPs alternately end with an I- and a P-frame.

For each layer, there are 16 distinct frames per GoP. The frame rate is equal to 25 frames per second — hence a GoP corresponds to $0.64s$ — and the spatial dimensions are equal to 720×304 pixels for the base layer as well as for the enhancement layer. Encoder quality parameters (QP, MeQP1 to MeQP5 and MeQPLP) are set to 31 and 25 for the base layer and the enhancement layer respectively. Further information on H.264/SVC encoding can be retrieved from [3].

As already mentioned, we investigate the H.264/SVC traces at the traffic class level. In terms of scalability, this means that we focus on modelling a fixed set of decodable substreams. Per GoP, there are 32 frames — 16 of the base and 16 of the enhancement layer — which we partition in the following 4 traffic classes.

- class 1: I and P frames of the base layer.
- class 2: B frames of the base layer.
- class 3: I and P frames of the quality enhancement layer.
- class 4: B frames of the quality enhancement layer.

Note that these traffic classes allow us to study the traces at temporal levels 0 and 4, with or without quality enhancement layer. The aggregate traffic of the following traffic classes constitute valid substreams: class 1, classes 1 and 2, classes 1 to 3, classes 1 to 4 and class 1, 2 and 4.

Although we focus on the traffic classes defined above in the remainder of this contribution, our approach can be applied for different sets of traffic classes or for multivariate traces stemming from other types of sources. From the vantage point of the characterisation algorithm under consideration, a trace is a sequence of row vectors $X = \{X_k, k = 0, \ldots, N-1\}$ where X_k is a row vector of size C whose jth element $X_{k,j}$ equals the number of bytes of class j that are generated during GoP k. Here, N

denotes the total number of GoPs in the trace and C denotes the number of traffic classes.

Statistics

The algorithm under consideration is based on matching various statistics of trace and model. The sample mean (row) vector and the $C \times C$ sample covariance matrix of the trace are defined as

$$\bar{X} = \frac{1}{N} \sum_{n=0}^{N-1} X_n \quad \text{and} \quad \text{cov}(X) = \frac{1}{N} \sum_{n=0}^{N-1} X_n' X_n - \bar{X}' \bar{X} \tag{1}$$

respectively. Here, primed vectors denote the transposed vectors as usual. Notice that we use the biased estimator of the covariance matrix. This will hardly influence the results since the numbers of GoPs N in the traces under consideration are sufficiently large.

With the trace X we associate the sum traces $S^{(k)}(X) = \{S_n^{(k)}(X), n = 0, \ldots, N-1\}$, with,

$$S_n^{(k)}(X) = \sum_{i=n}^{(n+k) \bmod N} X_i \,. \tag{2}$$

for $k = 0, 1, 2, \ldots$. Further, the (matrix valued) sample covariance function $\text{cf}(k)$ maps $k = 0, 1, 2, \ldots$ onto the sample covariance matrix of $S^{(k)}(X)$,

$$\text{cf}(k) = \frac{1}{N} \sum_{n=0}^{N-1} S_n^{(k)}(X)' S_n^{(k)}(X) - (k+1)^2 \bar{X}' \bar{X} \,. \tag{3}$$

Of particular interest for the statistical characterisation of the video source at the traffic class level are the scalar valued covariance functions of the aggregated traffic of a subset of the traffic classes. Let $\mathcal{C}$ be any subset of the set of traffic classes $\{1, 2, \ldots, C\}$ and let $X_{\mathcal{C}}$ denote the series

$$X_{\mathcal{C}} = \left\{ \sum_{j \in \mathcal{C}} X_{k,j}, k = 0, 1, \ldots N-1 \right\} \,. \tag{4}$$

The corresponding scalar valued covariance function $\text{cf}_{\mathcal{C}}(k)$ then relates to the matrix valued covariance function $\text{cf}(k)$ as follows,

$$\text{cf}_{\mathcal{C}}(k) = \epsilon_{\mathcal{C}} \, \text{cf}(k) \epsilon_{\mathcal{C}}' \,, \tag{5}$$

with $\epsilon_{\mathcal{C}}$ a row vector of size C whose ith element equals 1 if i is an element of $\mathcal{C}$ and 0 if this is not the case.

3 The multi-class DBMAP

Having introduced the traces and their statistics, we now focus on the Markov model for these traces. Let $q = \{q_i, i = 0, 1, \ldots\}$ denote an irreducible Markov chain defined on a finite state space $\mathcal{K} = \{1, \ldots, K\}$. Such a Markov chain is completely characterised by the transition probabilities $\gamma_{i,j}$, $i, j \in \mathcal{K}$. Further, let $\Gamma = [\gamma_{i,j}]_{i,j\in\mathcal{K}}$ denote the $K \times K$ transition matrix of the Markov chain and let $\pi = [\pi_1, \ldots, \pi_K]$ denote the row vector of its steady state probabilities. The vector π is the unique normalised solution of $\pi = \pi\Gamma$. The Markov chain q is referred to as the modulating Markov chain of the Discrete Batch-Markovian Arrival Process (DBMAP).

A multi-class DBMAP $x = \{x_k, k = 0, 1, 2, \ldots\}$ is a discrete time stochastic process which takes values in $\mathbb{R}^C$ such that the multivariate distribution of x_k only depends on the states q_k and q_{k+1} of the modulating Markov chain (for all k) [21]. Hence, this process is completely characterised by the transition matrix Γ of the modulating Markov chain and the doubly indexed set of multivariate distribution functions $\{\Phi_{i,j}(y), i, j \in \mathcal{K}, y \in \mathbb{R}^C\}$. $\Phi_{i,j}(y) = \Pr[x_{k,1} < y_1, \ldots x_{k,C} < y_C | q_k = i, q_{k+1} = j]$ is the multivariate distribution of x_k given $q_k = i$ and $q_{k+1} = j$. Here $x_{k,j}$ and y_j are the jth entries of the vectors x_k and y respectively.

Notice that in the literature DBMAPs only take values in $\mathbb{N}_+^C$. The process described here is somewhat more general but we will also refer to it as a DBMAP. Of course, one may discretise the traffic model for performance evaluation purposes. The resulting process is then a DBMAP as described in the literature.

In the remainder, we make some additional assumptions. Namely, (i) the process x_k at time k only depends on the state q_k and not on the transition $q_k \to q_{k+1}$ and (ii) for a given state q_k, the random vector x_k is multivariate normally distributed. Hence, the process (x_k, q_k) is completely characterised by the transition matrix Γ and the following mean vectors and covariance matrices in the different states $i \in \mathcal{K}$,

$$\mu_i = \mathrm{E}[x_k | q_k = i]\,, \quad \Omega_i = \mathrm{E}[x_k' x_k | q_k = i] - \mu_i' \mu_i\,. \tag{6}$$

Neither of these assumptions limit the applicability of our approach. It is well known that for every DBMAP there exists a DBMAP with the same characteristics such that the distributions only depend on the state and not on the transitions (although this comes at the cost of a larger state space) [23]. Further, the Gaussian assumption is not crucial for our approach. The methodology easily extends to non-Gaussian distributions as shown further. This assumption however allows for a fair comparison with the expectation-maximisation approach in Section 5. Moreover, the numerical results indicate that the Gaussian assumption does not impact the accuracy of the models in predicting buffer performance. Finally, notice that the normal distribution takes negative values with a positive probability. This probability will however turn out to be negligibly small for the DBMAPs that are generated by the characterisation algorithms.

Characteristics

We now retrieve some characteristics of the process x in terms of the characteristics Γ, π, μ_i and Ω_i of the multi-class DBMAP. The mean value vector μ and covariance matrix Ω are given by,

$$\mu = \mathrm{E}[x_k] = \sum_{i\in\mathcal{K}} \pi_i \mu_i\,, \qquad \Omega = \mathrm{E}[(x_k - \mu)'(x_k - \mu)] = \sum_{i\in\mathcal{K}} \pi_i(\Omega_i + \mu_i'\mu_i) - \mu'\mu\,. \tag{7}$$

Further, let $s^{(k)} = \{s_n^{(k)}, n = 0, 1, 2, \ldots\}$ denote the sum process corresponding to x. Here $s_n^{(k)}$ is defined as,

$$s_n^{(k)} = \sum_{i=n}^{n+k} x_i \,. \tag{8}$$

The matrix valued covariance function $\Theta(k)$ of the DBMAP then maps k on the covariance matrix of the process $s^{(k)}$,

$$\Theta(k) = \mathrm{E}[(s_n(k) - \mathrm{E}[s_n(k)])'(s_n(k) - \mathrm{E}[s_n(k)])] \,. \tag{9}$$

We can recursively calculate $\Theta(k)$ as follows. Since we have $s_n(k) = s_n(k-1) + x_{n+k}$ and by conditioning on the states of the modulating Markov chain, one shows that the covariance function is obtained recursively by the following expressions,

$$\Theta(k) = \Theta(k-1) + \Psi(k) + \Psi(k)' \,, \quad \Theta(0) = \Omega \,, \tag{10}$$

with,

$$\Psi(k) = \sum_{i,j \in \mathcal{K}} \psi_i(k-1)' \mu_j \gamma_{i,j} \,, \tag{11}$$

and,

$$\psi_l(k) = \mu_l + \sum_{j \in \mathcal{K}} \psi_j(k-1)\gamma_{j,l} \,, \quad \psi_l(0) = \mu_l \pi_l \,, \tag{12}$$

for $k = 1, 2, \ldots$ and for $l \in \mathcal{K}$.

As for the traces, the scalar valued covariance functions of the aggregated traffic of subsets of the traffic classes can be expressed in terms of the covariance function $\Theta(k)$. For any subset $\mathcal{C}$ of the traffic classes, let $x_{\mathcal{C}}$ denote following scalar valued DBMAP,

$$x_{\mathcal{C}} = \left\{ \sum_{j \in \mathcal{C}} x_{k,j}, k = 0, 1, 2, \ldots \right\} \,. \tag{13}$$

The scalar valued covariance function of $x_{\mathcal{C}}$ then equals,

$$\Theta_{\mathcal{C}}(k) = \epsilon_{\mathcal{C}} \Theta(k) \epsilon_{\mathcal{C}}' \,. \tag{14}$$

Recall that $\epsilon_{\mathcal{C}}$ denotes a row vector of size C whose ith element equals 1 if i is an element of $\mathcal{C}$ and 0 if this is not the case.

4 Genetic characterisation

With the statistics of the trace and the characteristics of the model established, we now introduce the genetic algorithm. The genetic approach under consideration exploits the fact that Markovian characterisation is almost trivial if not only the trace, but also the states of the modulating Markov chain are known. That is, given the trace X and a sequence of states $Q = \{Q_k, k = 0, \ldots, N-1\}$ associated to the consecutive GoPs, one

easily constructs a multi-class DBMAP by counting state transitions and by calculating conditional moments. We find the following DBMAP characteristics,

$$\begin{aligned} N_i &= \sum_{k=0}^{N-1} 1(Q_k = i)\,, \\ \gamma_{i,j} &= \frac{1}{N_i} \sum_{k=0}^{N-1} 1(Q_k = i, Q_{(k+1) \bmod N} = j)\,, \\ \mu_i &= \frac{1}{N_i} \sum_{k=0}^{N-1} X_k 1(Q_k = i)\,, \\ \Omega_i &= \frac{1}{N_i} \sum_{k=0}^{N-1} X_k' X_k 1(Q_k = i) - \mu_i' \mu_i\,. \end{aligned} \tag{15}$$

In the former expressions $1(\cdot)$ denotes the standard indicator function which evaluates to 1 if its argument is true and to 0 if this not the case. For counting the state transitions, it is assumed that the trace is played in a loop. That is, we also count the transition from sample $N-1$ to sample 0. Under this assumption, it is easy to show that the mean vector and the covariance matrix of the DBMAP with parameters as given in equation (15) match the sample mean and sample covariance matrix of the trace.

As mentioned before, we aim to find a model whose statistical properties closely match the corresponding statistics of the trace. In view of the introduction of the state assignment sequence Q, finding a multi-class DBMAP corresponds to finding a sequence Q such that the statistical properties of the derived DBMAP closely match the statistical properties of the trace. Sample mean and sample variance are matched automatically. To capture temporal correlation, the authors of [22] match the autocorrelation function of the DBMAP with the autocorrelation function of the trace. However, the autocorrelation of a DBMAP depends on the covariance matrix Ω of the DBMAP but not on the distinct covariance matrices Ω_i corresponding to the different states of the DBMAP. Hence, since it is our objective to accurately capture the correlation between the different traffic classes, the autocovariance function is of no use. Instead, we here focus on matching the sample covariance function of the trace with the covariance function of the DBMAP. In accordance with equations (10) to (12), the covariance function does depend on the matrices Ω_i.

Given the length of the trace N and the number of states of the Markov chain K, there are K^N possible sequences Q of state assignments. For the traces under consideration and for a Markov state space of size $K = 3$ this means that there are more than 10^{6000} possible sequences Q of state assignments. The size of the set of possible solutions clearly makes it infeasible to find the best solution by testing all possible solutions. We therefore use a genetic algorithm to search the space of all possible state assignments more efficiently.

Genetic algorithm

A genetic algorithm is a search and optimisation procedure inspired by Darwin's theory of evolution, see e.g. [24–26] for introductions on genetic algorithms. A genetic

algorithm transforms a set of candidate solutions into a new set of solutions using the Darwinian principle of reproduction and survival of the fittest. Candidate solutions are referred to as chromosomes: a generation of chromosomes is transformed into a new generation by means of mutation and crossover of chromosomes of the given generation (reproduction). Moreover, only the best chromosomes are retained (survival of the fittest).

In the present context, a chromosome is a sequence Q of state assignments to the consecutive GoPs. Its fitness measures the accuracy of the match of the covariance function of the DBMAP with the sample covariance function of the trace. In particular, the fitness of a chromosome Q is defined as follows,

$$\text{fitness}(Q) = \left(\frac{1}{L} \sum_{k=1}^{L} |w(\text{cf}(k) - \Theta(k))w'| \right)^{-1} . \tag{16}$$

Here $\text{cf}(k)$ is the sample covariance function of the trace as given in equation (3), L is the maximal lag for which the covariance function of the DBMAP and the sample covariance function of the trace are compared and $\Theta(k)$ is the covariance function of the DBMAP. The latter can be obtained by means of equations (10) to (12) in terms of the vectors π and μ_i $(i = 1, \ldots, K)$ and the matrices Γ and Ω_i $(i = 1, \ldots, K)$ that characterise the multi-class DBMAP. For a given chromosome Q, these vectors and matrices are given in equation (15). Finally, w is a weight row vector. Notice that a high fitness indicates an accurate match of covariance and sample covariance functions.

The row vector w in equation (16) allows one to assign different weights to the various elements of the covariance matrix. In particular, one can assign weights to the variance functions of aggregated traffic from subsets of the traffic classes as follows. For all i, let α_i denote the weight assigned to the variance function of the aggregated traffic class $\mathcal{C}_i$ ($\mathcal{C}_i \subseteq \{1, 2, \ldots, C\}$ is a subset of the set of traffic classes). In view of property (5) of sample covariance functions and in view of property (14) of the covariance function of a DBMAP, we obtain the following weight vector,

$$w = \sum_i \sqrt{\alpha_i}\, \epsilon_{\mathcal{C}_i} . \tag{17}$$

Recall that $\epsilon_{\mathcal{C}_i}$ is a row vector of size C whose ith element equals 1 if i is an element of $\mathcal{C}$ and 0 if this is not the case. An example weight vector is determined in Section 5.

Given the fitness function, the execution of the genetic algorithm can be summarised as follows. First, a random generation of G chromosomes is created. Each chromosome is created by randomly assigning states to the consecutive GoPs. Then, the following substeps are performed until a sufficiently fit chromosome is found.

1. Calculate the fitness of each chromosome in the current generation. The fitness of the generation is defined as the fitness of the fittest chromosome of the generation.
2. Create a new generation of $G - 1$ chromosomes by the following operations:
   (a) Mutation: randomly choose a chromosome of the current generation with probability inversely proportional to the fitness of the chromosome. Then either (i) change a random continuous block of state assignments to a randomly chosen state or (ii) reverse a random continuous block of state assignments. Add this new (mutated) chromosome to the new generation.

(b) Crossover: randomly choose two chromosomes of the current generation with probability inversely proportional to the fitness of these chromosomes. Interchange the state assignments of a randomly chosen sub-part of the state assignments. Add the fittest of these new chromosomes to the new generation.

3. Copy the fittest chromosome of the current generation to the new generation.

Notice that step 3 implies that the fitness of consecutive generations never decreases. The genetic characterisation algorithm is numerically evaluated in the next Section.

5 Numerical results

We now evaluate the genetic video characterisation approach by some numerical examples. We here focus on the characterisation of the trace "Brotherhood of the Wolf". Qualitatively similar results were obtained for the trace "Pirates of the Caribbean" but are omitted for brevity. In addition, similar results were also obtained for these traces with the GoPs partitioned into different sets of traffic classes.

For all examples, the fitness of a chromosome is based on the (sample) covariance functions of lag $L = 1$ up to lag $L = 10$ and a generation consists of $G = 10$ chromosomes. We assign the same weight to all decodable substreams: class 1, class 1 and 2, class 1 and 3, class 1 to 3 and all classes constitute decodable substreams. Hence, the weight vector is set to

$$w = [5\,3\,3\,1] = [1\,0\,0\,0] + [1\,1\,0\,0] + [1\,0\,1\,0] + [1\,1\,1\,0] + [1\,1\,1\,1]\,.$$

To assess the accuracy of our approach, we benchmark our genetic approach against the well-known maximum likelihood approach. This approach optimises the parameters of a stochastic model such that the likelihood that the model generates the trace is maximal. In particular, the maximum likelihood problem under consideration is a maximum likelihood problem with incomplete data in the sense that only the sample points of the trace but not the Markov states of the modulating chain are known. A well known optimisation algorithm for this type of problems is the expectation maximisation (EM) algorithm [27]. More specifically, Liporace's EM algorithm is used [28]. This is a Baum-Welch type algorithm (see a.o. Baum et al. [29] and Welch [30]) for multivariate hidden Markov models. Liporace assumes that the multivariate distributions associated with the different states belong to a certain class of multivariate continuous distributions which includes the multivariate normal distribution, hence its applicability to the DBMAP characterisation at hand. This algorithm suffers some arithmetic underflow issues which are most easily addressed by rescaling, see [31]. The interested reader is referred to the references above for more information on the EM algorithm.

Expectation Maximisation is a local likelihood maximisation algorithm. Let $\Upsilon = \{\Lambda, \mu_i, \Omega_i, i \in \mathcal{K}\}$ denote the parameter set of the DBMAP. Given an initial parameter set Υ_0, the EM algorithm recalculates this parameter set, thereby converging to a set which locally maximises the likelihood. To avoid local maxima, we have applied random restart. In the examples below, we have repeated the EM algorithm 100 times and retained the parameter set corresponding to the highest likelihood. The initial parameter set Υ_0 is chosen randomly as follows:

– The transition matrix is obtained by normalising the rows of a matrix whose entries are independent random variables, uniformly distributed in $(0, 1]$.

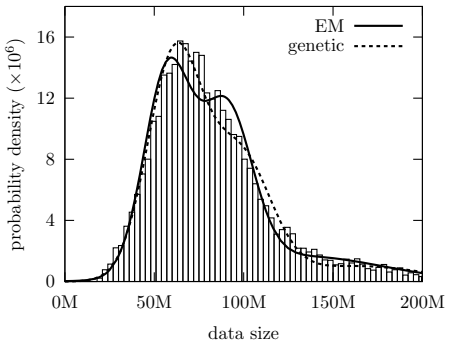

Fig. 2 Histogram and probability density functions for aggregated data of class 1 and 2 and chains with 3 states.

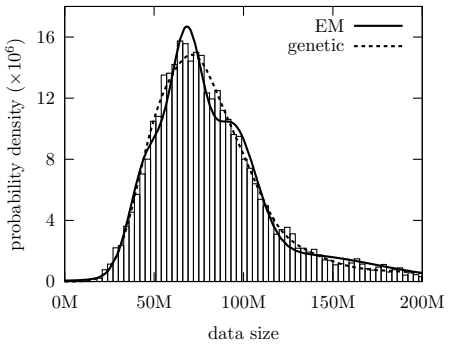

Fig. 3 Histogram and probability density functions for aggregated data of class 1 and 2 and chains with 5 states.

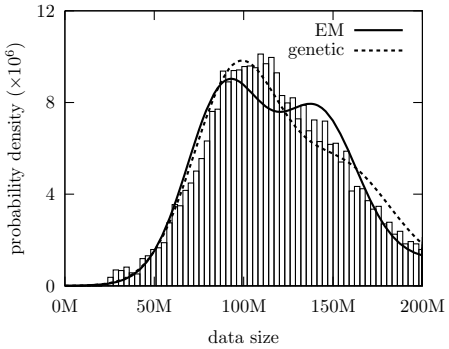

Fig. 4 Histogram and probability density functions for aggregated data of all classes and chains with 3 states.

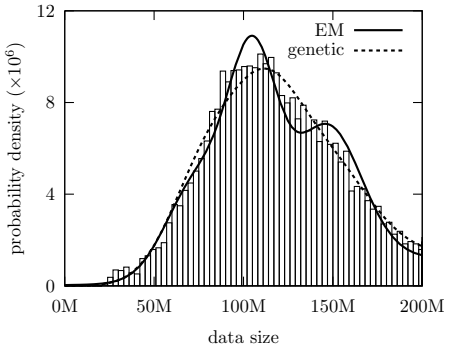

Fig. 5 Histogram and probability density functions for aggregated data of all classes and chains with 5 states.

– For each i in $\mathcal{K}$, let $\mathcal{U}_i$ and $\hat{\mathcal{U}}_i$ denote diagonal matrices whose diagonal entries are independent random variables, uniformly distributed in $(0, 2]$. We then set $\mu_i = \bar{X}\,\mathcal{U}_i$ and $\Omega_i = \hat{\mathcal{U}}_i\,\mathrm{cov}(X)\,\hat{\mathcal{U}}_i$.

In Figures 2 to 5, the density functions of the models are compared with the histogram of the trace. Note that the histogram is rescaled by the bin size to facilitate comparison. Figures 2 and 3 show the density functions and histograms of the aggregated data of class 1 and 2 and for DBMAPs with 3 and 5 states respectively. In these figures as well as in the following ones, we always benchmark our genetic approach against the EM approach. With 3 states and for either approach, we already note a reasonable visual correspondence between histogram and density functions, despite the limited size of the state space. Extending the state space to 5 states visually yields a better match for both approaches. This is also confirmed by calculation of the Cramèr-von Mises distance d_{CvM} and Kolmogorov distance d_{K} between the cumulative distribution function of the DBMAPs and the empirical cumulative distribution function [32], see Table 1. These distances are defined as follows,

$$d_{\mathrm{CvM}} = \int_{-\infty}^{\infty} (F_n(x) - F(x))^2 dF(x)\,, \quad d_{\mathrm{K}} = \sup_x |F_n(x) - F(x)|\,,$$

(a) Cramèr-von Mises distance

	genetic			EM		
	3	*5*	*10*	*3*	*5*	*10*
class 1–2	0.009364	0.0033198	0.0018828	0.011228	0.0059986	0.0035118
class 1–4	0.011575	0.0052250	0.0028004	0.012861	0.0061486	0.0036272

(b) Kolmogorov distance

	genetic			EM		
	3	*5*	*10*	*3*	*5*	*10*
class 1–2	0.016521	0.0081553	0.0051796	0.021887	0.0131525	0.0073455
class 1–4	0.020549	0.0132269	0.0078226	0.023441	0.0136937	0.0070517

Table 1 Distance between empirical distribution functions and distribution functions of the models

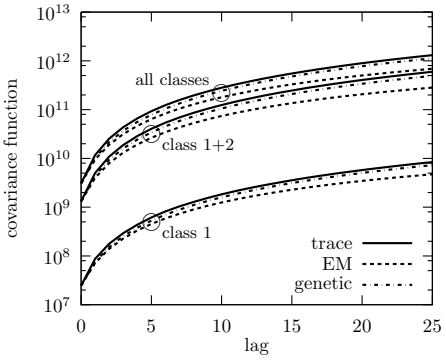

Fig. 6 Covariance function of model and trace for chains with 3 states

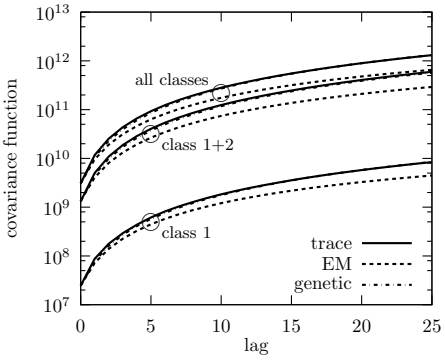

Fig. 7 Covariance function of model and trace for chains with 5 states

with $F(x)$ and $F_n(x)$ the distribution function of the DBMAP and the empirical distribution function of the trace, respectively. As can be seen from this table, the distance decreases upon adding states for both approaches. This is not unexpected for EM. The maximum likelihood principle of the EM approach takes into account the complete distribution function in each state. Since DBMAPs with more states allow for more flexibility, a closer match can be found. For the genetic optimisation procedure, the improvement is somewhat unexpected. The genetic approach does not take into account the complete distribution; optimisation is based on the first two moments in each state. Nevertheless, the distance (Cramèr-von Mises and Kolmogorov) between distribution and empirical distribution decreases by adding more states and is even somewhat smaller than EM. Figures 4 and 5 show histogram and density functions of the aggregated data of all classes, again for chains with 3 and 5 states. The corresponding distance between distribution and empirical distribution are also tabulated in Table 1. Also in this case, more states yield better correspondence between histogram and density functions for both approaches. Moreover, the match further improves by adding even more states as can be seen from Table 1 where the distance is also tabulated for 10 state DBMAPs. Other results, not shown here, confirm that the distance decreases for EM for DBMAPs with increasing state space. However, the distance may or may not decrease for the genetic approach.

In Figures 6 and 7 the covariance functions of various decodable subtraces are depicted for the models generated by the genetic and the EM approach. As opposed to the quality of the match of the histograms, the genetic approach clearly outperforms the EM approach. As figure 6 shows, 3 states are not sufficient to visually match the variance function of the trace with either approach. However, by extending the state space to 5 states, the variance function of the genetic approach is visually almost indistinguishable from the variance function of the trace. Results not shown here confirm that extending the state space more yields an even better match. In all cases, the genetic approach outperforms the EM approach.

Although a close match of various statistics is desirable, the accuracy of the models in predicting queueing behaviour is the main objective of traffic characterisation. To assess the accuracy of the DBMAP models of H.264/SVC video for performance evaluation, we compare queueing performance measures by means of trace and DBMAP based simulations. In particular, the queueing system under consideration is a discrete-time finite capacity first-in-first-out multi-server queueing system with partial buffer sharing (PBS). The slot time of the discrete-time queueing system is set to the length of a GoP and all traffic stemming from the different video sources is packetised into fixed length packets of 1500 bytes. Different sources are multiplexed before entering the queue. Given the bandwidth BW of the output line in $Mbit/s$ and the GoP length g in ms, one easily calculates the number of packets P that can leave the queueing system during a slot, $P = \lceil BW\, g/12 \rceil$. In the remainder, all performance measures are obtained by simulating the PBS queue over 10^8 GoPs. For the trace based simulations, the same trace is played over and over. However, a new offset value is randomly chosen every time the trace finishes.

In figures 8 to 10 the probability mass functions of the queue content of the different traffic classes are depicted. Both results from trace based simulations and from model based simulations are depicted. The models are either generated by the genetic approach or by the EM approach, as indicated. The bandwidth of the output line equals $8Mbit/s$ and the queue can store up to 200 packets, excluding the packets that leave the queue during the slot. Class 2, 3 and 4 packets are accepted as long as the buffer is less than 90%, 80% and 70% occupied respectively. Input traffic comes from 4 sources, either from the trace or the model.

The genetic approach yields a reasonable match which improves by increasing the size of the state space. Visually, the match is already quite accurate for 5 states while the 10-state model predicts the density function of the queue size very accurately. In contrast, only for the DBMAP with 10 states, we visually get a good match by the EM approach (except for the class 2 packets where the match remains poor). Further, notice that the model based probability mass function of most traffic classes takes smaller values than the trace based probability mass function. This implies that the DBMAP models overestimate the probability that there are no packets (of the class under consideration) in the buffer and underestimate the packet loss probability.

Finally, in Figure 11, the packet loss ratios (PLR) of the different traffic classes are depicted versus the number of sources (every source is the trace or model of "Brotherhood of the Wolf") that are multiplexed in the PBS buffer. Table 2 also displays the absolute values (abs) of the PLR of models and trace as well as the relative values (rel) of the models with respect to the value of the trace for the simulation with 4 sources. The output line of the PBS queue has a fixed bandwidth of $12Mbit/s$ and DBMAPs with 10 states are considered. Again, the genetic approach outperforms the EM approach which is in correspondence with the observations for the probability mass

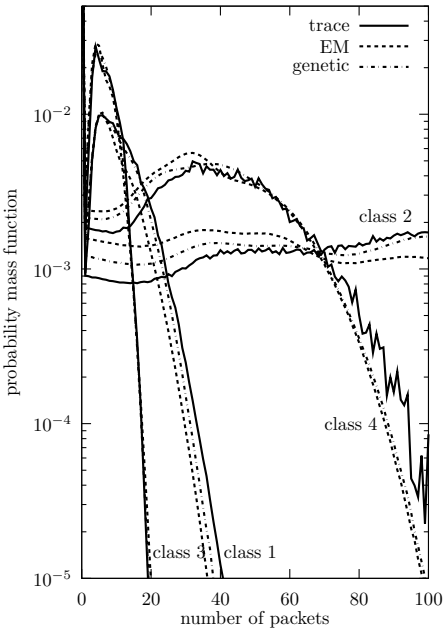

Fig. 8 Probability mass function of the queue content of the different classes (3 states).

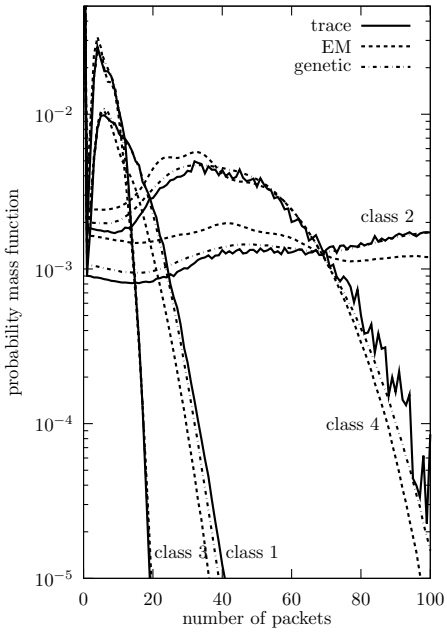

Fig. 9 Probability mass function of the queue content of the different classes (5 states).

	trace	EM		genetic	
	abs	*abs*	*rel*	*abs*	*rel*
class 1	1.38e-7	4.66e-8	0.338	9.11e-8	0.660
class 2	5.50e-4	1.55e-4	0.282	4.39e-4	0.798
class 3	1.64e-3	5.61e-4	0.342	1.39e-3	0.848
class 4	3.49e-3	1.37e-3	0.392	2.96e-3	0.848

Table 2 Packet loss ratio of the different classes (10 states) and 4 sources.

functions. It is well known that the decay rate of the probability mass function of the queue content of a queue is a good estimator for the decay rate of the packet loss ratio (as a function of the queue size) [33]. Hence, an accurate assessment of the packet loss ratio requires an accurate match of the probability mass function.

6 Conclusions and extensions

In this contribution we proposed multi-class DBMAPs to characterise H.264/SVC scalable video at the sub-GoP level. We showed that — by means of a genetic algorithm — multi-class DBMAPs with a limited state space can be found that accurately capture the characteristics of the video traces. In comparison with the EM approach, the models can more accurately predict performance measures of the traces in various queueing scenarios. In general, the size of the state space of the DBMAP will be key for obtaining accurate results. In all cases, adding states yield more accurate results. However, if the model is to be combined with a numerical queueing analysis, a larger

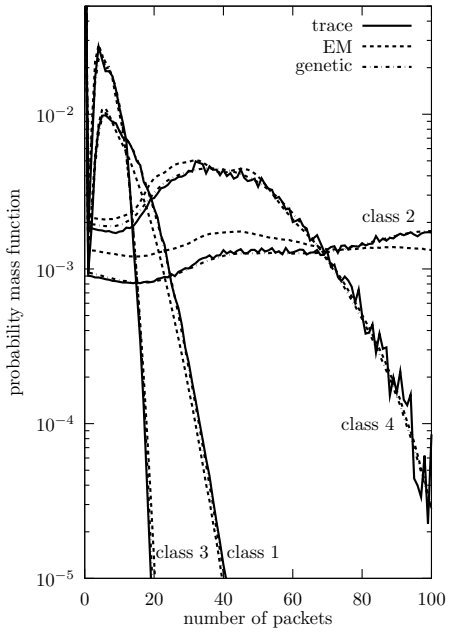

Fig. 10 Probability mass function of the queue content of the different classes (10 states).

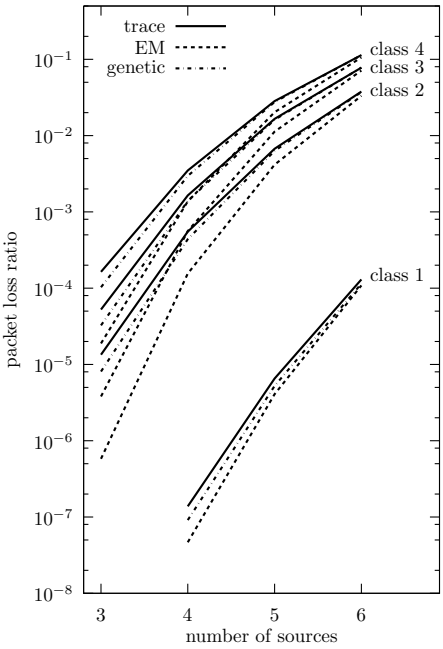

Fig. 11 Packet loss ratio of the different classes (10 states).

state space results in an increase of the computational effort and there is a trade-off between accuracy and computational effort.

We conclude with some possible extensions of the methodology presented here. In the paper, the restriction is imposed that all conditional distribution functions Φ_i are multivariate normal distributions. This restriction allows one to limit the number of parameters of the DBMAP but comes at the cost of a less accurate match of the histograms and density functions as shown in Figures 2 and 3. Since the genetic approach results in a series Q of state assignments to the consecutive samples of the trace, one easily retrieves "conditional" histograms of the sample points of the trace corresponding to a particular state. Given these conditional histograms, a more accurate description of these conditional distribution functions may be possible. The most accurate description is obtained when the conditional histograms are used as conditional density functions. In this particular case, a perfect match of density functions and histograms is obtained.

The fact that the genetic approach results in a series Q of state assignments to the consecutive samples of the trace, also allows us to investigate the trace at the frame level or at any other sub-GoP level. Let Z_k denote the vector whose different elements correspond to the sizes of the different frames of the kth GoP and let Z denote the sequence $\{Z_k, k = 0, \ldots, N-1\}$. In view of the GoP structure displayed in figure 1, every element Z_k of the series Z is a vector of size 32. Taking into account the correlation between the different frames, it is reasonable to assume that the temporal correlation of the trace can be captured by investigating the temporal correlation of a limited number of traffic classes. Hence, the series X can be used to characterise temporal correlation which results in a series Q of state assignments. The series Q and

Z can then be combined to obtain a 32 dimensional DBMAP which characterises the trace at the frame level. The same reasoning can be repeated to characterise the trace at any other sub-GoP level.

Acknowledgement

This work has been carried out in the framework of the Q-MATCH project sponsored by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). The first author is a Postdoctoral Fellow with the Research Foundation, Flanders (F.W.O.-Vlaanderen), Belgium. A preliminary version of this paper was presented at the 15th International Conference on Analytical and Stochastic Modelling Techniques and Applications, Nicosia, Cyprus, 3–6 June 2008.

References

1. H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen. Scalable internet video using MPEG-4. *Signal Processing: Image communication*, 15(1-2):95–126, 1999.
2. P. Seeling, M. Reisslein, and B. Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial. *IEEE Communications Surveys*, 6(3):58–78, 2004.
3. H. Schwartz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9): 1103–1120, 2007.
4. N. W. Macfadyen. Traffic characterisation and modeling. *BT Technology Journal*, 20(3):14–30, 2002.
5. J. Liebeherr and D.E. Wrege. Traffic characterization algorithms for VBR video in multimedia networks. *Multimedia Systems*, 6:271–283, 1998.
6. M. R. Izquierdo and D. S. Reeves. A survey of statistical source models for variable-bit-rate compressed video. *Multimedia Systems*, 7(3):199–213, May 1999.
7. M. Conti, E. Gregori, and A. Larsson. Study of the impact of MPEG-1 correlations on video-sources statistical multiplexing. *IEEE Journal of Selected Areas in Communications*, 14(7):1455–1471, 1996.
8. A. Lombardo, G. Morabito, and G. Schembra. An accurate and treatable Markov model of MPEG-video traffic. In *Proceedings of 17th Conference on Computer Communications (INFOCOM 1998)*, pages 217–224, San Francisco, California, 1998.
9. M. Conti and E. Gregori. Modeling MPEG scalable sources. *Multimedia Tools and Applications*, 13(2):127–145, February 2001.
10. D. Moltchanov, Y. Koucheryavy, and J. Harju. The model of single smoothed MPEG traffic source based on the D-BMAP arrival process with limited state space. In *Proceedings of ICACT 2003*, pages 57–63, Phoenix Park, South Korea, 2003.
11. J Zhao, B Li, and I Ahmad. Traffic model for layered video: an approach on Markovian arrival process. In *Packet Video 2003*, Nantes, France, 2003.
12. S. Baey. Modeling MPEG4 video traffic based on a customization of the DBMAP. In *Proceedings of SPECTS 2004*, 2004.
13. Q.T. Zhang. A general AR-based technique for the generation of arbitrary gamma VBR video traffic in ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1130–1137, 1999.
14. A. A. Alheraish and S. A. Alshebeili. AR-based quadratic modeling for GOP MPEG-encoded video traffic in ATM networks. *Computer Communications*, 27(1):81–92, January 2004.
15. Y. Won and S. Ahn. GOP ARIMA: Modeling the nonstationarity of VBR processes. *Multimedia Systems*, 10(5):359–378, August 2005.
16. A. Lazaris, P. Koutsakis, and M. Paterakis. On modeling video traffic from multiplexed MPEG-4 videoconference streams. *Next Generation Teletraffic And Wired/ Wireless Advanced Networking, Proceedings*, 4003:46–57, 2006.

17. C. H. Liew, C. K. Kodikara, and A. M. Kondoz. MPEG-encoded variable bit-rate video traffic modelling. *IEE Proceedings-Communications*, 152(5):749–756, October 2005.
18. N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao. On modeling MPEG video traffics. *IEEE Transactions On Broadcasting*, 48(4):337–347, December 2002.
19. X. D. Huang, Y. H. Zhou, and R. F. Zhang. A multiscale model for MPEG-4 varied bit rate video traffic. *IEEE Transactions On Broadcasting*, 50(3):323–334, September 2004.
20. G. Latouche and V. Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*. Series on statistics and applied probability. ASA-SIAM, 1999.
21. C. Blondia and O. Casals. Statistical multiplexing of VBR sources: A matrix-analytic approach. *Performance Evaluation*, 16(1–3):5–20, 1992.
22. S. Kempken and W. Luther. Modeling of H.264 high definition video traffic using discrete-time semi-Markov processes. In *Managing Traffic Performance in Converged Networks, Proceedings of the 20th International Teletraffic Congress (Lecture Notes in Computer Science 4516)*, 2007.
23. G. Hasslinger. Semi-Markovian modelling and performance analysis of variable rate traffic in ATM networks. *Telecommunication Systems*, 7(1–3):281–298, 1997.
24. J.R. Koza. Survey of genetic algorithms and genetic programming. In *Proceedings of Wescon 1995*, San Francisco, CA, 1995.
25. M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996, ISBN 0262133164.
26. S.N. Sivanandam, and S.N. Deepa, *Introduction to genetic algorithms*. Springer Berlin Heidelberg, 2008, ISBN 978-3-540-73189-4.
27. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1976.
28. L. A. Liporace. Maximum likelihood estimation for mulitvariate observation of Markov sources. *IEEE Transactions on Information Theory*, 28(5):729–733, 1982.
29. L. E. Baum, T. Petrie, and N. Soules, G. Weis. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.
30. L. R. Welch. Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):1,10–13, 2003.
31. L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
32. J. Jurecková, P. Kumar Sen. *Robust Statistical Procedures: Asymptotics and Interrelations*. Wiley series in probability and mathematical statistics, 1996, ISBN: 978-0-471-82221-9.
33. H. Kim and N. Shroff. Loss probability calculations and asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Transactions on Networking*, 9(6):755–768, 2001.